

AmpliFi Holder Dividend Engine Master Specifications & Build Documentation

This document defines the complete technical, architectural, and operational specifications for transforming the existing AmpliFi platform into a simplified Holder Dividend Engine, while explicitly preserving all current working utilities required for the rework. This document is designed to be directly consumable by GPT-5.2 or senior engineers for deterministic implementation.

1. Strategic Objective

The goal of this rework is to reduce AmpliFi to a single, high-signal utility: automatic, weighted weekly payouts to the top 50 longest and largest token holders via a claim-based system. All unnecessary engagement, social scanning, and campaign logic will be removed or disabled, while preserving wallet auth, reward vaults, claim systems, and security primitives.

Key Principles

- 1 Minimal surface area – fewer moving parts, fewer attack vectors.
- 2 Deterministic payouts – predictable, auditable logic.
- 3 No utility regression – existing working systems are reused, not rewritten.
- 4 Upgradeable distribution logic – future-proof without refactor.

2. System Components to Preserve

- 1 Wallet authentication (Phantom / Solana standard adapters).
- 2 Reward vault / treasury signer architecture.
- 3 Claim window smart validation (nonce / epoch based).
- 4 Existing payout execution & transaction batching utilities.
- 5 Admin controls for funding, pausing, and emergency halts.

3. Components to Remove or Disable

- 1 Twitter / social ingestion pipelines.
- 2 Engagement scoring and point attribution.
- 3 Campaign configuration and creator-engager routing.
- 4 Social verification layers.

4. New Core Architecture

The new architecture centers on a periodic holder snapshot, ranking engine, weighted distribution calculator, and claim execution layer.

- 1 Snapshot Service – weekly capture of holder balances and timestamps.
- 2 Ranking Engine – deterministic selection of top 50 holders.

- 3 Distribution Engine – weighted reward calculation.
- 4 Claim Window – user-initiated reward claiming.

5. Holder Snapshot System

A scheduled job runs once per week to capture the holder state.

- 1 Fetch all token holders via RPC indexer.
- 2 Track wallet address, balance, and first-seen timestamp.
- 3 Persist snapshots with epoch ID.

Suggested schema:

wallet | balance | first_seen | last_snapshot | epoch_id

6. Ranking Logic

Ranking is executed immediately after snapshot finalization.

- 1 Exclude wallets that sold during the epoch (optional toggle).
- 2 Primary sort: holding duration.
- 3 Secondary sort: token balance.
- 4 Select top 50 wallets.

7. Distribution Formula

Each wallet receives a weighted share of the weekly creator reward pool.

Recommended formula:

$\text{weight} = (\text{holding_duration_days} ^ 0.6) \times (\text{balance} ^ 0.4)$

Weights are normalized across the top 50 to determine payout percentages.

8. Claim Window Mechanics

- 1 Claim window opens once per week after ranking finalization.
- 2 Wallet must match snapshot + epoch ID.
- 3 Single claim per wallet enforced.
- 4 Unclaimed funds handled via configurable policy.

9. Admin Controls

- 1 Fund weekly reward vault.
- 2 Trigger snapshot manually if needed.
- 3 Open / close claim window.
- 4 Emergency pause / kill switch.

10. Frontend Pages

- 1 Landing – explanation of holding incentives.
- 2 Leaderboard – top 50 holders.
- 3 Claim Page – allocation, timer, claim button.
- 4 Admin Panel – restricted access.

11. Security Considerations

- 1 Epoch-based nonces to prevent replay.
- 2 One signer vault model.
- 3 Double-entry accounting for payouts.
- 4 Full on-chain transaction logging.

12. Build Order (Execution Plan)

- 1 Disable social & campaign modules.
- 2 Implement snapshot cron.
- 3 Add ranking + weighting logic.
- 4 Wire distribution to existing claim system.
- 5 Finalize frontend trimming.
- 6 Security review & dry-run epoch.

13. Final Outcome

The resulting system is a clean, auditable Holder Dividend Engine that preserves AmpliFi's strongest infrastructure while removing all unnecessary complexity. This rework positions the platform as a universal, token-agnostic dividend rail.